

Sparse Signal Estimation by Maximally Sparse Convex Optimization

Ivan W. Selesnick and Ilker Bayram

Abstract—This paper addresses the problem of sparsity penalized least squares for applications in sparse signal processing, e.g. sparse deconvolution. This paper aims to induce sparsity more strongly than L1 norm regularization, while avoiding non-convex optimization. For this purpose, this paper describes the design and use of non-convex penalty functions (regularizers) constrained so as to ensure the convexity of the total cost function, F , to be minimized. The method is based on parametric penalty functions, the parameters of which are constrained to ensure convexity of F . It is shown that optimal parameters can be obtained by semidefinite programming (SDP). This maximally sparse convex (MSC) approach yields maximally non-convex sparsity-inducing penalty functions constrained such that the total cost function, F , is convex. It is demonstrated that iterative MSC (IMSC) can yield solutions substantially more sparse than the standard convex sparsity-inducing approach, i.e., L1 norm minimization.

I. INTRODUCTION

In sparse signal processing, the ℓ_1 norm has special significance [4], [5]. It is the convex proxy for sparsity. Given the relative ease with which convex problems can be reliably solved, the ℓ_1 norm is a basic tool in sparse signal processing. However, penalty functions that promote sparsity more strongly than the ℓ_1 norm yield more accurate results in many sparse signal estimation/reconstruction problems. Hence, numerous algorithms have been devised to solve non-convex formulations of the sparse signal estimation problem. In the non-convex case, generally only a local optimal solution can be ensured; hence the initialization and the specification of algorithm parameters become important, as does the particular choice of non-convex penalty function.

This paper aims to develop an approach that promotes sparsity more strongly than the ℓ_1 norm, but which attempts to avoid, as far as possible, complications arising in non-convex optimization. In particular, the paper addresses ill-posed linear inverse problems of the form

$$\arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ F(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sum_{n=0}^{N-1} \lambda_n \phi_n(x_n) \right\} \quad (1)$$

where $\lambda_n > 0$ and $\phi_n : \mathbb{R} \rightarrow \mathbb{R}$ are sparsity-inducing regularizers (penalty functions) for $n \in \mathbb{Z}_N = \{0, \dots, N-1\}$.

I. W. Selesnick is with the Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, 6 Metrotech Center, Brooklyn, NY 11201, USA. Email: selesi@poly.edu, phone: 718 260-3416, fax: 718 260-3906. I. Bayram is with the Department of Electronics and Communication Engineering, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey. Email: ilker.bayram@itu.edu.tr.

This research was supported by the NSF under Grant No. CCF-1018020. February 26, 2013

Problems of this form arise in denoising, deconvolution, compressed sensing, etc.

This paper explores the use of non-convex penalty functions ϕ_n , under the constraint that the total cost function F is convex and therefore reliably minimized. To this end, we employ penalty functions parameterized by variables a_n , i.e., $\phi_n(x) = \phi(x; a_n)$, wherein the parameters a_n are selected so as to ensure convexity of the total cost function F . A key idea of the proposed approach is that the parameters a_n can be optimized to make the penalty functions ϕ_n maximally non-convex (i.e., maximally sparsity-inducing), subject to the constraint that F is convex. We refer to this as the ‘maximally-sparse convex’ (MSC) approach. The allowed interval for the parameters a_n , to ensure F is convex, is obtained by formulating a semidefinite program (SDP) [2], which is itself a convex optimization problem. Hence, in the proposed MSC approach, the cost function F to be minimized depends itself on the solution to a convex problem. This paper also describes an iterative MSC (IMSC) approach that boosts the applicability and effectiveness of the MSC approach.

The proposed MSC approach requires a suitable parametric penalty function $\phi(\cdot; a)$, where a controls the degree to which ϕ is non-convex. Therefore, this paper also addresses the choice of parameterized non-convex penalty functions so as to enable the approach. The paper proposes suitable penalty functions ϕ and describes their relevant properties.

A. Related Work (Threshold Functions)

When \mathbf{H} in (1) is the identity operator, the problem is one of denoising and is separable in x_n . In this case, a sparse solution \mathbf{x} is usually obtained by some type of threshold function, $\theta : \mathbb{R} \rightarrow \mathbb{R}$. The simplest and perhaps most widely used threshold functions are the soft and hard threshold functions [19]. Each has its disadvantages, and many other thresholding functions that provide a compromise of the soft and hard thresholding functions have been proposed – for example: the firm threshold [28], the non-negative (nn) garrote [24], [27], the SCAD threshold function [22], [61], and the proximity operator of the non-convex ℓ_p quasi-norm, i.e., $\phi(x) = |x|^p$, $0 < p < 1$ [39]. Several penalty functions are unified by the two-parameter formulas given in [3], [30], wherein threshold functions are derived as proximity operators [17]. (Table 1.2 of [17] lists the proximity operators of numerous functions.) Further threshold functions are defined directly by their functional form [58]–[60].

The threshold function used in this paper is developed as a proximity operator (i.e., via a penalty function, ϕ , or rather,

via the derivative of a penalty function, ϕ'), and is designed so as to have the three properties advocated in [22]: unbiasedness (of large coefficients), sparsity, and continuity. Further, the threshold function θ and its corresponding penalty function ϕ are parameterized by two parameters: the threshold T and the right-sided derivative of θ at the threshold, i.e. $\theta'(T^+)$, a measure of the threshold function's sensitivity. Like other threshold functions, the proposed threshold function biases large $|x_n|$ less than does the soft threshold function, but is continuous unlike the hard threshold function. As will be shown below, the proposed function is most similar to the threshold function (proximity operator) corresponding to the logarithmic penalty, but it is designed to have less bias. It is also particularly convenient in algorithms for solving (1) that do not call on the threshold function directly, but instead call on the derivative of penalty function, $\phi'(x)$, due to its simple functional form. Such algorithms include iterative reweighted least squares (IRLS) [33], iterative reweighted ℓ_1 [10], [57], FOCUSS [48], and algorithms derived using majorization-minimization (MM) [23] wherein the penalty function is upper bounded (e.g. by a quadratic or linear function).

Sparsity-based nonlinear estimation algorithms can also be developed by formulating suitable non-Gaussian probability models that reflect sparse behavior, and by applying Bayesian estimation techniques [1], [15], [21], [34], [35], [43], [45]. A novel approach in this vein is based on codelength minimization [47]. We note that, the approach we take below is essentially a deterministic one; we do not explore its formulation from a Bayesian perspective.

B. Related Work (Sparsity Penalized Least Squares)

Numerous problem formulations and algorithms to obtain sparse solutions to the general ill-posed linear inverse problem, (1), have been proposed. The ℓ_1 norm penalty (i.e., $\phi_n(x) = |x|$) has been proposed for sparse deconvolution [9], [14], [36], [54] and more generally for sparse signal processing [13] and statistics [55]. For the ℓ_1 norm and other non-differentiable convex penalties, efficient algorithms for large scale problems of the form (1) and similar (including convex constraints) have been developed based on proximal splitting methods [16], [17], alternating direction method of multipliers (ADMM) [8], majorization-minimization (MM) [23], primal-dual gradient descent [20], and Bregman iterations [31].

Several approaches aim to obtain solutions to (1) that are more sparse than the ℓ_1 norm solution. Some of these methods proceed first by selecting a non-convex penalty function that induces sparsity more strongly than the ℓ_1 norm, and second by developing non-convex optimization algorithms for the minimization of F ; for example, iterative reweighted least squares (IRLS) [33], [57], FOCUSS [32], [48], and [11], [42], [53]. With the availability of fast reliable algorithms for ℓ_1 norm minimization, reweighted ℓ_1 norm minimization is a suitable approach for the non-convex problem [10], [57]: the tighter upper bound of the non-convex penalty provided by the weighted ℓ_1 norm, as compared to the weighted ℓ_2 norm, reduces the chance of convergence to poor local minima. Other algorithmic approaches include ‘difference of convex’ (DC) programming [29] and operator splitting [12].

A distinct approach to obtain sparse solutions to (1) is to find an approximate solution minimizing the ℓ_0 quasi-norm or satisfying an ℓ_0 constraint. Examples of such algorithms include: matching pursuit (MP) and orthogonal MP (OMP) [40], greedy ℓ_1 [38], iterative hard thresholding (IHT) [6], [7], [37], [44], hard thresholding pursuit [26], smoothed ℓ_0 , [41], iterative support detection (ISD) [56], single best replacement (SBR) [51], and ECME thresholding [46].

In contrast to these works, in this paper the penalties ϕ_n are constrained by the operator \mathbf{H} . This approach (MSC) deviates from the usual approach which chooses the penalty not based on the system (or ‘channel’) \mathbf{H} through which \mathbf{x} is observed, but on prior knowledge of \mathbf{x} . We also note that, by design, the proposed approach leads to a convex optimization problem; hence, it differs from approaches that pursue non-convex optimization. It also differs from usual convex approaches for sparse signal estimation/recovery which utilize convex penalties. In this paper, the aim is precisely to utilize non-convex penalties that induce sparsity more strongly than a convex penalty possibly can.

Compared to algorithms aiming to solve the ℓ_0 quasi-norm problem, the proposed approach again differs. First, the ℓ_0 problem is highly non-convex, while the proposed approach defines a convex problem. Second, methods for ℓ_0 seek the correct support (index set of non-zero elements) of \mathbf{x} and do not regularize (penalize) any element x_n in the calculated support.¹ In contrast, the design of the regularizer (penalty) is at the center of the proposed approach, and no x_n is left unregularized.

Once a reasonably sparse solution is obtained, by convex ℓ_1 norm minimization or otherwise, it can be helpful to subsequently perform least squares over only the non-zero elements of the obtained solution. This is referred to as ‘debiasing’ [25], and can be used as an optional post-processing step for the proposed MSC approach as well.

II. SCALAR THRESHOLD FUNCTIONS

The proposed threshold function and corresponding penalty function is intended to serve as a compromise between soft and hard threshold functions, and as a parameterized family of functions for use with the proposed MSC method for ill-posed linear inverse problems, to be described in Sect. III.

First, we note the high sensitivity of the hard threshold function to small changes in its input. If the input is slightly less than the threshold T , then a small positive perturbation produces a large change in the output, i.e., $\phi_h(T - \epsilon) = 0$ and $\phi_h(T + \epsilon) \approx T$ where $\phi_h : \mathbb{R} \rightarrow \mathbb{R}$ denotes the hard threshold function. Due to this discontinuity, spurious noise peaks/bursts often appear as a result of hard-thresholding denoising. For this reason, a continuous threshold function is often preferred. The susceptibility of a threshold function θ to the phenomenon of spurious noise peaks can be roughly quantified by the maximum value its derivative attains, i.e., $\max_{y \in \mathbb{R}} \theta'(y)$, provided θ is continuous. For the threshold functions considered below, θ' attains its maximum values

¹Some ℓ_0 methods, e.g. reweighted ℓ_1 techniques, utilize regularization within an algorithm, but the final result is not regularized over its support.

at $y = \pm T^+$; hence, the value of $\theta'(T^+)$ will be noted. The soft threshold function θ_s has $\theta'_s(T^+) = 1$ reflecting its insensitivity. However, θ_s substantially biases (attenuates) large values of its input; i.e., $\theta_s(y) = y - T$ for $y > T$.

A. Problem Statement

In this section, we seek a threshold function and corresponding penalty (i) for which the ‘sensitivity’ $\theta'(T^+)$ can be readily tuned from 1 to infinity and (ii) that does not substantially bias large y , i.e., $y - \theta(y)$ decays to zero rapidly as y increases.

For a given penalty function ϕ , the proximity operator [17] denoted $\theta : \mathbb{R} \rightarrow \mathbb{R}$ is defined by

$$\theta(y) = \arg \min_{x \in \mathbb{R}} \left\{ F(x) = \frac{1}{2}(y - x)^2 + \lambda \phi(x) \right\} \quad (2)$$

where $\lambda > 0$. Common sparsity-inducing penalties include

$$\phi(x) = |x| \quad \text{and} \quad \phi(x) = \frac{1}{a} \log(1 + a|x|). \quad (3)$$

We similarly assume in the following that $\phi(x)$ is differentiable for all $x \in \mathbb{R}$ except $x = 0$, and that ϕ is symmetric, i.e., $\phi(-x) = \phi(x)$.

If $\theta(y) = 0$ for all $|y| \leq T$ for some $T > 0$, and T is the maximum such value, then the function θ is a threshold function and T is the threshold.

It is often beneficial in practice if θ admits a simple functional form. However, as noted above, a number of algorithms for solving (1) do not use θ directly, but use ϕ' instead. In that case, it is beneficial if ϕ' has a simple function form. This is relevant in Sec. III where such algorithms will be used.

In order that $y - \theta(y)$ approaches zero, the penalty function ϕ must be non-convex, as shown by the following.

Proposition 1. Suppose $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a convex function and $\theta(y)$ denotes the proximity operator associated with ϕ , defined in (2). If $0 \leq y_1 \leq y_2$, then

$$y_1 - \theta(y_1) \leq y_2 - \theta(y_2). \quad (4)$$

Proof: Let $u_i = \theta(y_i)$ for $i = 1, 2$. We have,

$$y_i \in u_i + \lambda \partial \phi(u_i). \quad (5)$$

Since $y_2 \geq y_1$, by the monotonicity of both of the terms on the right hand side of (5), it follows that $u_2 \geq u_1$.

If $u_2 = u_1$, (4) holds with since $y_2 \geq y_1$.

Suppose now that $u_2 > u_1$. Note that the subdifferential $\partial \phi$ is also a monotone mapping since ϕ is a convex function. Therefore it follows that if $z_i \in \lambda \partial \phi(u_i)$, we should have $z_2 \geq z_1$. Since $y_i - \theta(y_i) \in \lambda \partial \phi(u_i)$, the claim follows. ■

According to the proposition, if the penalty is convex, then the gap between $\theta(y)$ and y increases as the magnitude of y increases. The larger y is, the greater the bias (attenuation) is. The soft threshold function is an extreme case that keeps this gap constant (beyond the threshold T , the gap is equal to T). Hence, in order to avoid attenuation of large values, the penalty function must be non-convex.

B. Properties

Suppose F , defined in (2), is convex and $\phi(x)$ is differentiable for all $x \in \mathbb{R}$ except $x = 0$. Then the subdifferential ∂F is given by

$$\partial F(x) = \begin{cases} \{x - y + \lambda \phi'(x)\}, & \text{if } x \neq 0, \\ [\lambda \phi'(0^-), \lambda \phi'(0^+)] - y, & \text{if } x = 0. \end{cases} \quad (6)$$

Since F is convex, its minimizer x^* satisfies $0 \in \partial F(x^*)$. From (6), if $y \in [\lambda \phi'(0^-), \lambda \phi'(0^+)]$, then $0 \in \partial F(0)$, and in turn $x^* = 0$. Assuming ϕ is symmetric, $\phi'(0^-) = -\phi'(0^+)$, this interval represents the thresholding interval of θ , and the threshold T is given by

$$T = \lambda \phi'(0^+). \quad (7)$$

Suppose now that $y \notin [\lambda \phi'(0^-), \lambda \phi'(0^+)]$. This can happen if either of the following hold: (i) $y > \lambda \phi'(0^+)$, (ii) $y < \lambda \phi'(0^-)$. In the sequel, we will study the case (i). The results can be extended to (ii) straightforwardly.

First, note that if $y > \lambda \phi'(0^+)$, then $x^* > 0$ and it satisfies

$$y = x^* + \lambda \phi'(x^*). \quad (8)$$

Let us define $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ as

$$f(x) = x + \lambda \phi'(x). \quad (9)$$

For $y > \lambda \phi'(0^+)$, the threshold function θ can now be expressed as

$$\theta(y) = f^{-1}(y). \quad (10)$$

Observe that f is continuous and $f(0^+) = \lambda \phi'(0^+) = T$. In view of (10), this implies that $\theta(T^+) = 0$. Thus, $\theta(y)$ is continuous at the threshold.

For a symmetric ϕ , it can be shown that F is convex if and only if f is monotone, i.e., $f'(x) > 0, \forall x > 0$. This in turn is equivalent to

$$\phi''(x) > -\frac{1}{\lambda}, \quad \forall x > 0. \quad (11)$$

Let us now find the first and second derivatives of $\theta(y)$ at $y = T^+$. From (10),

$$f(\theta(y)) = y. \quad (12)$$

Differentiating with respect to y gives

$$f'(\theta(y)) \theta'(y) = 1. \quad (13)$$

Differentiating again with respect to y gives

$$f''(\theta(y)) [\theta'(y)]^2 + f'(\theta(y)) \theta''(y) = 0. \quad (14)$$

Setting $y = T^+$ in (13) gives

$$\theta'(T^+) = \frac{1}{f'(0^+)}. \quad (15)$$

Setting $y = T^+$ in (14) gives

$$f''(0^+) [\theta'(T^+)]^2 + f'(0^+) \theta''(T^+) = 0 \quad (16)$$

or

$$\theta''(T^+) = -\frac{f''(0^+)}{[f'(0^+)]^3}. \quad (17)$$

Using (9), we have

$$f'(0^+) = 1 + \lambda\phi''(0^+) \quad \text{and} \quad f''(0^+) = \lambda\phi'''(0^+).$$

Hence

$$\theta'(T^+) = \frac{1}{1 + \lambda\phi''(0^+)} \quad (18)$$

and

$$\theta''(T^+) = -\frac{\lambda\phi'''(0^+)}{[1 + \lambda\phi''(0^+)]^3}. \quad (19)$$

Equations (18) and (19) will be used in the following. As noted above, $\theta'(T^+)$ reflects the maximum sensitivity of θ . The value $\theta''(T^+)$ is also relevant; it will be set in Sec. II-D so as to induce $\theta(y) - y$ to decay rapidly to zero.

C. The Logarithmic Penalty Function

The logarithmic penalty function can be used for the MSC method to be described in Sec. III. It also serves as the model for the penalty function developed in Sec. II-D below, designed to have less bias. The logarithmic penalty is given by

$$\phi(x) = \frac{1}{a} \log(1 + a|x|), \quad 0 < a \leq \frac{1}{\lambda} \quad (20)$$

which is differentiable except at $x = 0$. For $x \neq 0$, the derivative of ϕ is given by

$$\phi'(x) = \frac{1}{1 + a|x|} \text{sign}(x), \quad x \neq 0, \quad (21)$$

as illustrated in Fig. 1a. The function $f(x) = x + \lambda\phi'(x)$ is illustrated in Fig. 1b. The threshold function θ , given by (10), is illustrated in Fig. 1c.

Let us find the range of a for which F is convex. Note that

$$\phi''(x) = -\frac{a}{(1 + ax)^2} \quad \text{for } x > 0 \quad (22)$$

$$\phi'''(x) = \frac{2a^2}{(1 + ax)^3} \quad \text{for } x > 0. \quad (23)$$

Using the condition (11), it is deduced that if $0 \leq a \leq 1/\lambda$, then $f(x)$ is increasing, the cost function F in (2) is convex, and the threshold function θ is continuous.

The threshold is given by $T = \lambda$. To find $\theta'(T^+)$ and $\theta''(T^+)$, note that

$$\phi''(0^+) = -a, \quad \phi'''(0^+) = 2a^2. \quad (24)$$

Using (18) and (19), we have

$$\theta'(T^+) = \frac{1}{1 - a\lambda}, \quad \theta''(T^+) = -\frac{2a^2\lambda}{(1 - a\lambda)^3}. \quad (25)$$

As a varies between 0 and $1/\lambda$, the derivative $\theta'(T^+)$ varies between 1 and infinity. As a approaches zero, θ approaches the soft-threshold function. We can set a so as to specify $\theta'(T^+)$. Solving (25) for a gives

$$a = \frac{1}{\lambda} \left(1 - \frac{1}{\theta'(T^+)} \right). \quad (26)$$

Therefore, T and $\theta'(T^+)$ can be directly specified by setting the parameters λ and a (i.e., $\lambda = T$ and a is given by (26)).

Note that $\theta''(T^+)$ given in (25) is strictly negative except when $a = 0$ which corresponds to the soft threshold function.

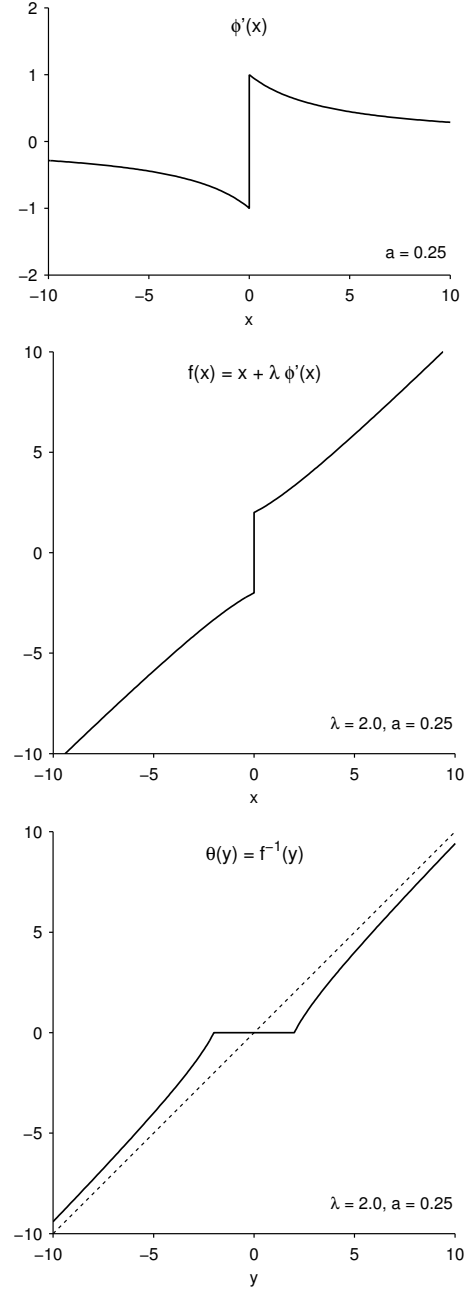


Fig. 1. Functions related to the logarithmic penalty function (20).

The negativity of $\theta''(T^+)$ inhibits the rapid approach of θ to the identity function.

The threshold function θ is obtained by solving $y = f(x)$ for x , leading to

$$ax^2 + (1 - ay)x + (\lambda - y) = 0, \quad (27)$$

which leads in turn to the explicit formula

$$\theta(y) = \begin{cases} \left[\frac{|y|}{2} - \frac{1}{2a} + \sqrt{\left(\frac{|y|}{2} + \frac{1}{2a}\right)^2 - \frac{\lambda}{a}} \right] \text{sign}(y), & |y| \geq \lambda \\ 0, & |y| \leq \lambda \end{cases}$$

as illustrated in Fig. 1c. As shown, the gap $y - \theta(y)$ goes to zero for large y . By increasing a up to $1/\lambda$, the gap goes to zero more rapidly; however, increasing a also changes $\theta'(T^+)$. The

single parameter a affects both the derivative at the threshold and convergence rate to identity.

The next section derives a penalty function, for which the gap goes to zero more rapidly, for the same value of $\theta'(T^+)$. It will be achieved by setting $\theta''(T^+) = 0$.

D. The Arctangent Penalty Function

To obtain a penalty approaching the identity more rapidly than the logarithmic penalty, we use equation (21) as a model, and define a new penalty by means of its derivative as

$$\phi'(x) = \frac{1}{bx^2 + a|x| + 1} \text{sign}(x), \quad a > 0, b > 0. \quad (28)$$

Using (7), the corresponding threshold function θ has threshold $T = \lambda$. In order to use (18) and (19), we note

$$\phi''(x) = -\frac{(2bx + a)}{(bx^2 + ax + 1)^2} \quad \text{for } x > 0$$

$$\phi'''(x) = \frac{2(2bx + a)^2}{(bx^2 + ax + 1)^3} - \frac{2b}{(bx^2 + ax + 1)^2} \quad \text{for } x > 0.$$

The derivatives at zero are given by

$$\phi'(0^+) = 1, \quad \phi''(0^+) = -a, \quad \phi'''(0^+) = 2a^2 - 2b. \quad (29)$$

Using (18), (19), and (29), we have

$$\theta'(T^+) = \frac{1}{1 - \lambda a}, \quad \theta''(T^+) = \frac{2\lambda(b - a^2)}{(1 - \lambda a)^3}. \quad (30)$$

We may set a so as to specify $\theta'(T^+)$. Solving (30) for a gives (26), the same as for the logarithmic penalty function.

In order that the threshold function increases rapidly toward the identity function, we use the parameter b . To this end, we set b so that θ is approximately linear in the vicinity of the threshold. Setting $\theta''(T^+) = 0$ in (30) gives $b = a^2$. Therefore, the proposed penalty function is given by

$$\phi'(x) = \frac{1}{a^2x^2 + a|x| + 1} \text{sign}(x). \quad (31)$$

From the condition (11), we find that if $0 \leq a \leq 1/\lambda$, then $f(x) = x + \lambda\phi'(x)$ is increasing, F is convex, and θ is continuous. The parameters a and λ , can be set as for the logarithmic penalty function; namely $T = \lambda$ and by (26).

To find the threshold function θ , we solve $y = x + \lambda\phi'(x)$ for x which leads to

$$a^2x^3 + a(1 - |y|a)x^2 + (1 - |y|a)x + (\lambda - |y|) = 0 \quad (32)$$

for $|y| > T$. The value of $\theta(y)$ can be found solving the cubic polynomial for x , and multiplying the real root by $\text{sign}(y)$. Although θ does not have a simple functional form, the function ϕ' does. Therefore, algorithms such as MM and IRLS, which use ϕ' instead of θ , can be readily used in conjunction with this penalty function.

The penalty function itself, ϕ , can be found by integrating its derivative:

$$\phi(x) = \int_0^{|x|} \phi'(u) du \quad (33)$$

$$= \frac{2}{a\sqrt{3}} \left(\tan^{-1} \left(\frac{1 + 2a|x|}{\sqrt{3}} \right) - \frac{\pi}{6} \right). \quad (34)$$

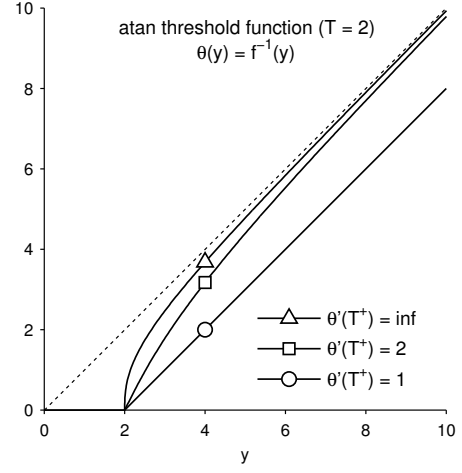


Fig. 2. The arctangent threshold function for several values of $\theta'(T^+)$.

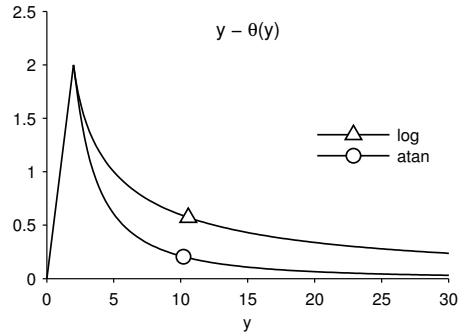
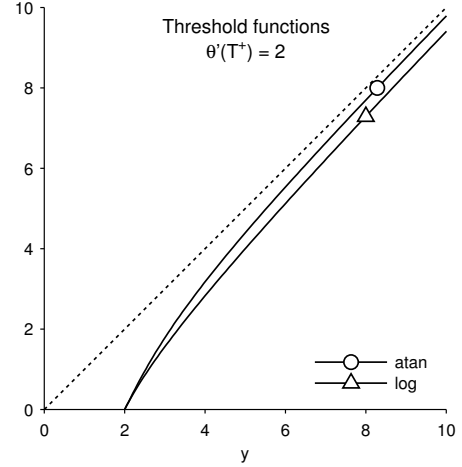


Fig. 3. Comparison of arctangent and logarithmic penalty functions, both with $\theta'(T^+) = 2$. The arctangent threshold function approaches the identity faster than the logarithmic penalty function.

We refer to this as the arctangent penalty function.

The threshold function is illustrated in Fig. 2 for threshold $T = \lambda = 2$ and three values of $\theta'(T^+)$. With $\lambda = 2$, the function F is convex for all $a \in [0, 1/\lambda]$. With $\theta'(T^+) = 1$, one gets $a = 0$ and θ is the soft-threshold function. With $\theta'(T^+) = 2$, one gets $a = 1/4$ and θ converges to the identity function. With $\theta'(T^+) = \infty$, one gets $a = 1/2$; in this case, θ converges more rapidly to the identity function, but θ may be more sensitive than desired in the vicinity of the threshold.

Figure 3 compares the logarithmic and arctangent threshold functions where the parameters for each function are set so that T and $\theta'(T^+)$ are the same (specifically, $\lambda = T = 2$ and $a = 1/4$). It can be seen that the arctangent threshold function converges more rapidly to the identity function than the logarithmic threshold function. To illustrate the difference more clearly, the lower panel in Fig. 3 shows the gap between the identity function and the threshold function. For the arctangent threshold function, this gap goes to zero more rapidly. Yet, for both threshold functions, θ' has a maximum value of 2. The faster convergence of the arctangent threshold function is due to $\phi'(x)$ going to zero like $1/x^2$, whereas for the logarithmic threshold function $\phi'(x)$ goes to zero like $1/x$.

E. Other Penalty Functions

The firm threshold function [28], and the smoothly clipped absolute deviation (SCAD) threshold function [22], [61] also provide a compromise of hard and soft thresholding. Moreover, both are continuous and equal to the identity function for large $|y|$. For both functions, $\phi'(x) = 0$ for large x . However, some forms of algorithms such as IRLS, MM, etc., involve dividing by ϕ' , which then leads to divide-by-zero issues.

A widely used penalty function is the ℓ_p pseudo-norm, $0 < p < 1$, for which $\phi(x) = |x|^p$. However, using (11), it can be seen that for this penalty function, the cost function $F(x)$ is not convex for any $0 < p < 1$. Therefore, θ has a discontinuity [39], like the hard threshold function, and is likewise sensitive to small changes in y . As our current interest is in non-convex penalty functions for which F is convex, we do not further discuss the ℓ_p penalty. The reader is referred to [39] for in-depth analysis of this penalty function.

F. Denoising Example

To illustrate the trade-off between $\theta'(T^+)$ and the bias introduced by thresholding, we consider the denoising of the noisy signal illustrated in Fig. 4. Wavelet domain thresholding is performed with several thresholding functions.

Each threshold function is applied with the same threshold, $T = 3\sigma$. Most of the noise (c.f. the ‘three-sigma rule’) will fall below the threshold and will be eliminated. The RMSE-optimal choice of threshold is usually lower than 3σ , so this represents a larger threshold than that usually used. However, a larger threshold reduces the number of spurious noise peaks produced by hard thresholding.

The hard threshold achieves the best RMSE, but the output signal exhibits spurious noise bursts due to noisy wavelet coefficients exceeding the threshold. The soft threshold function reduces the spurious noise bursts, but attenuates the peaks and results in a higher RMSE. The arctangent threshold function suppresses the noise bursts, with modest attenuation of peaks, and results in an RMSE closer to that of hard thresholding.

In this example, the signal is ‘bumps’ from WaveLab [18], with length 2048. The noise is additive white Gaussian noise with standard deviation $\sigma = 0.4$. The orthonormal Daubechies wavelet with 3 vanishing moments was used for this example.

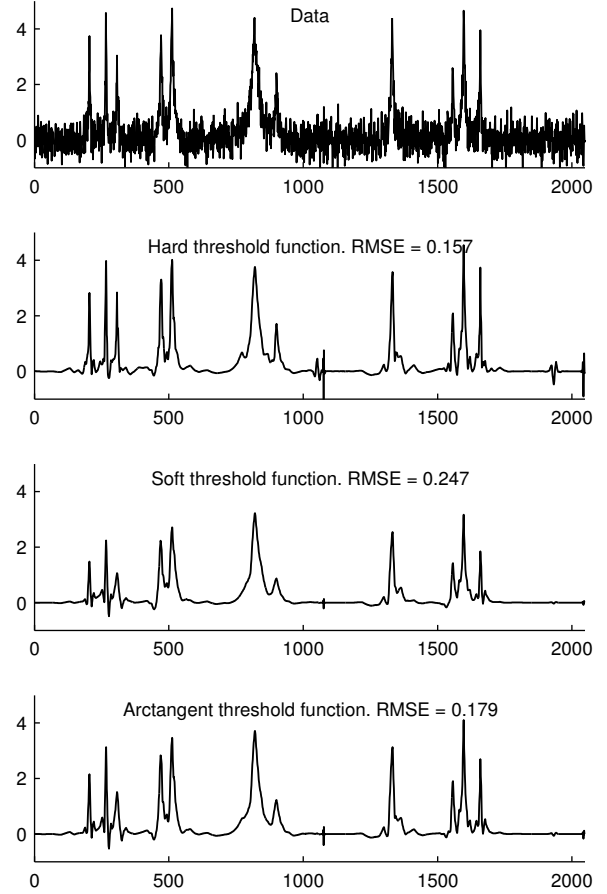


Fig. 4. Denoising via orthonormal wavelet thresholding using various threshold functions.

III. SPARSITY PENALIZED LEAST SQUARES

Consider the linear model,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (35)$$

where $\mathbf{x} \in \mathbb{R}^N$ is a sparse N -point signal, $\mathbf{y} \in \mathbb{R}^M$ is the observed signal, $\mathbf{H} \in \mathbb{R}^{M \times N}$ is linear operator (e.g., convolution), and $\mathbf{w} \in \mathbb{R}^M$ is additive white Gaussian noise (AWGN). The vector \mathbf{x} is denoted $\mathbf{x} = (x_0, \dots, x_{N-1})^T$.

Under the assumption that \mathbf{x} is sparse, we consider the linear inverse problem:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ F(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sum_{n=0}^{N-1} \lambda_n \phi(x_n; a_n) \right\} \quad (36)$$

where $\phi(x; a)$ is a sparsity-promoting penalty function with parameter a , such as the logarithmic or arctangent penalty functions. In many applications, all λ_n are equal, i.e., $\lambda_n = \lambda$. For generality, we let this regularization parameter depend on the index n .

In the following, we address the question of how to constrain the regularization parameters λ_n and a_n so as to ensure F is convex, even when $\phi(\cdot; a_n)$ is not convex.

A. Convexity Condition

Let $\phi(x; a)$ denote a penalty function with parameter a . Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$, defined as

$$f(x) = \frac{1}{2}x^2 + \lambda\phi(x; a). \quad (37)$$

Assume that, for special choices of λ and a , $f(x)$ can be made convex. Let us give a name to the set of all such special choices.

Definition 1. Let \mathcal{S} be the set of pairs (λ, a) for which $f(x)$ in (37) is convex. We refer to \mathcal{S} as the ‘parameter set associated with ϕ ’.

For the logarithmic and arctangent penalty functions described above, the set \mathcal{S} is given by

$$\mathcal{S} = \{(\lambda, a) : \lambda > 0, 0 \leq a \leq 1/\lambda\}. \quad (38)$$

Now, consider the function $F : \mathbb{R}^N \rightarrow \mathbb{R}$, defined in (36). The following proposition provides a sufficient condition on (λ_n, a_n) ensuring the convexity of F .

Proposition 2. Suppose \mathbf{R} is a positive semi-definite diagonal matrix such that $\mathbf{H}^T\mathbf{H} - \mathbf{R}$ is also positive semi-definite. Let r_n denote the n -th diagonal entry of \mathbf{R} , i.e., $[\mathbf{R}]_{n,n} = r_n \geq 0$. Also, let \mathcal{S} be the parameter set associated with ϕ . Then, $F(\mathbf{x})$ in (36) is convex if $(\lambda_n/r_n, a_n) \in \mathcal{S}$ for each n .

Proof: The function $F(\mathbf{x})$ can be written as

$$F(\mathbf{x}) = \underbrace{\frac{1}{2}\mathbf{x}^T(\mathbf{H}^T\mathbf{H} - \mathbf{R})\mathbf{x} - \mathbf{y}^T\mathbf{H}\mathbf{x} + \frac{1}{2}\mathbf{y}^T\mathbf{y}}_{q(\mathbf{x})} + g(\mathbf{x}), \quad (39)$$

where

$$g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{R}\mathbf{x} + \sum_n \lambda_n \phi(x_n; a_n). \quad (40)$$

Note that $q(\mathbf{x})$ is convex since $\mathbf{H}^T\mathbf{H} - \mathbf{R}$ is positive semi-definite. Now, since \mathbf{R} is diagonal, we can rewrite $g(\mathbf{x})$ as

$$g(\mathbf{x}) = \sum_n \frac{r_n}{2} |x_n|^2 + \lambda_n \phi(x_n; a_n) \quad (41)$$

$$= \sum_n r_n \left(\frac{1}{2} |x_n|^2 + \frac{\lambda_n}{r_n} \phi(x_n; a_n) \right), \quad (42)$$

From (42), it follows that if $(\lambda_n/r_n, a_n) \in \mathcal{S}$ for each n , then $g(\mathbf{x})$ is convex. Under this condition, being a sum of two functions, it also follows that $F(\mathbf{x})$ is convex. ■

The proposition states that constraints on the penalty parameters a_n ensuring convexity of $F(\mathbf{x})$ can be obtained using a diagonal matrix \mathbf{R} lower bounding $\mathbf{H}^T\mathbf{H}$. In view of (38), the following is a corollary of this result.

Corollary 1. For the logarithmic and arctangent penalty functions, if

$$0 \leq a_n \leq \frac{r_n}{\lambda_n}, \quad (43)$$

then $F(\mathbf{x})$ in (36) is convex. □

If $r_k = 0$ for some $k \in \mathbb{Z}_N$, then the corollary guarantees the convexity of F if $\phi(\cdot; a_k)$ is convex. For the logarithmic and arctangent penalty functions, that implies $a_k = 0$, and

$\phi(\cdot; a_k)$ reduces to the absolute value function. On the other hand, if $r_k > 0$ for some k , then F can be convex even though $\phi(\cdot; a_k)$ is not convex.

We illustrate condition (43) with a simple example using $N = 2$ variables. We set $\mathbf{H} = \mathbf{I}$, $\mathbf{y} = [9.5, 9.5]^T$, and $\lambda_0 = \lambda_1 = 10$. Then $\mathbf{R} = \mathbf{I}$ is a positive diagonal matrix with $\mathbf{H}^T\mathbf{H} - \mathbf{R}$ positive semidefinite. According to (43), F is convex if $a_i \leq 0.1$, $i = 0, 1$. Figure 5 illustrates the contours of the logarithmic penalty function and the cost function F for three values of a . For $a = 0$, the penalty function reduces to the ℓ_1 norm. Both the penalty function and F are convex. For $a = 0.1$, the penalty function is non-convex but F is convex. The non-convexity of the penalty is apparent in the figure (its contours do not enclose convex regions). The non-convex ‘star’ shaped contours induce sparsity more strongly than the diamond shaped contours of the ℓ_1 norm. For $a = 0.2$, both the penalty function and F are non-convex. The non-convexity of F is apparent in the figure (a convex function can not have more than one stationary point, while the figure shows two). In this case, the star shape is too pronounced for F to be convex. In this example, $a = 0.1$ yields the maximally sparse convex (MSC) problem.

How can an optimal \mathbf{R} be obtained? Let us denote the minimal eigenvalue of $\mathbf{H}^T\mathbf{H}$ by α_{\min} . Then $\mathbf{R} = \alpha_{\min}\mathbf{I}$ is a positive semi-definite diagonal lower bound, as needed. However, this is a sub-optimal lower bound in general. For example, if \mathbf{H} is a non-constant diagonal matrix, then an optimal lower bound is $\mathbf{H}^T\mathbf{H}$ itself, which is very different from $\alpha_{\min}\mathbf{I}$ in general. A tighter lower bound is of interest because the tighter the bound, the more non-convex the penalty function can be, while maintaining convexity of F . In turn, sparser solutions can be obtained without sacrificing convexity of the cost function. A tighter lower bound can be found as the solution to an optimization problem, as described in the following.

B. An Optimal Diagonal Lower Bound

Given \mathbf{H} , the convexity conditions above calls for a positive semidefinite diagonal matrix \mathbf{R} lower bounding $\mathbf{H}^T\mathbf{H}$. In order to find an optimal lower bound, each r_n should be maximized. However, these N parameters must be optimized jointly. We formulate the calculation of \mathbf{R} as an optimization problem:

$$\begin{aligned} & \arg \max_{\mathbf{r} \in \mathbb{R}^N} \sum_{n=0}^{N-1} r_n \\ & \text{such that } r_n \geq \alpha_{\min} \\ & \mathbf{H}^T\mathbf{H} - \mathbf{R} \geq 0 \end{aligned} \quad (44)$$

where \mathbf{R} is the diagonal matrix $[\mathbf{R}]_{n,n} = r_n$. The inequality $\mathbf{H}^T\mathbf{H} - \mathbf{R} \geq 0$ expresses the constraint that $\mathbf{H}^T\mathbf{H} - \mathbf{R}$ is positive semidefinite (all its eigenvalues non-negative). Note that the problem is feasible, because $\mathbf{R} = \alpha_{\min}\mathbf{I}$ satisfies the constraints.

Problem (44) can be recognized as a semidefinite optimization problem, a type of convex optimization problem for which algorithms have been developed and for which software is

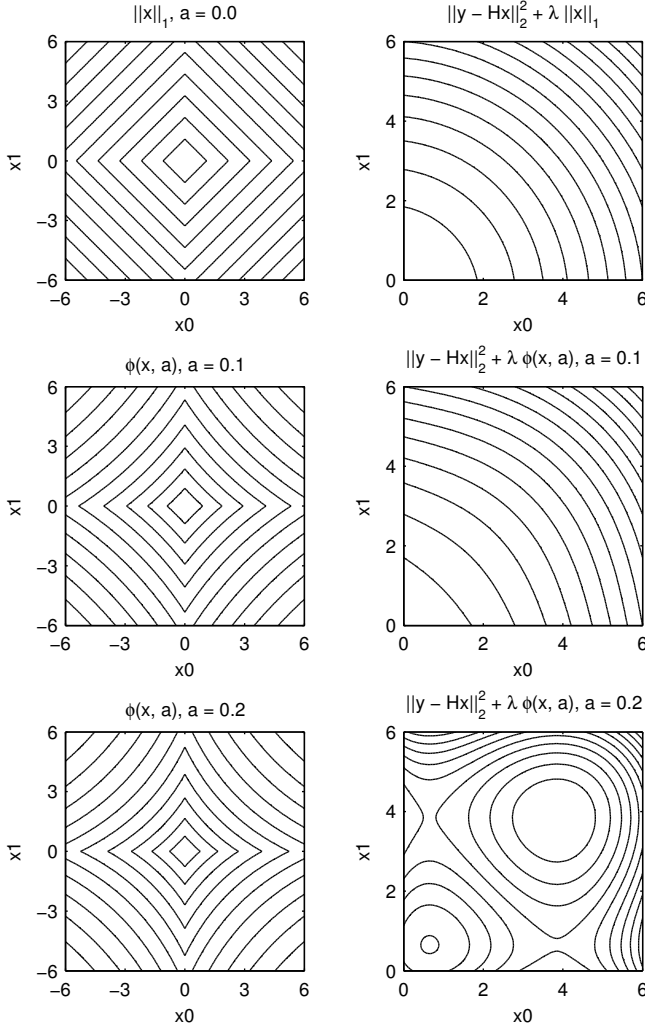


Fig. 5. Contour plots of the logarithmic penalty function ϕ and cost function F for three values of a as described in the text. For $a = 0.1$, the function F is convex even though the penalty function is not.

available [2]. The cost function in (44) is a linear function of the N variables, and the constraints are linear matrix inequalities (LMIs). To solve (44) and obtain an optimal \mathbf{R} , we have used the MATLAB software package ‘SeDuMi’ [52].

Often, inverse problems arising in signal processing involve large data sets (e.g., speech, EEG, and images). Practical algorithms must be efficient in terms of memory and computation. In particular, they should be ‘matrix-free’, i.e., the operator \mathbf{H} is not explicitly stored as a matrix, nor are individual rows or columns of \mathbf{H} accessed or modified. However, optimization algorithms for semidefinite programming usually involve row/column matrix operations and are not ‘matrix free’. Hence, solving problem (44) will likely be a bottleneck for large scale problems. This motivates the development of semidefinite algorithms for solving (44) where \mathbf{H} is not explicitly available, but for which multiplications by \mathbf{H} and \mathbf{H}^T are fast.

Nevertheless, for 1D problems of ‘medium’-size, (44) is readily solved via existing software. In case (44) is too computationally demanding, then the suboptimal choice $\mathbf{R} = \alpha_{\min} \mathbf{I}$

can be used. Furthermore, we describe below a multistage algorithm whereby the proposed MSC approach is applied iteratively.

C. Optimality Conditions and Threshold Selection

When the cost function F in (36) is convex, then its minimizer must satisfy specific conditions [4, Prop 1.3]. These conditions can be used to verify the optimality of a solution produced by a numerical algorithm. The conditions also aid in setting the regularization parameters λ_n .

If F in (36) is convex, and ϕ is differentiable except at zero, then \mathbf{x}^* minimizes F if

$$\begin{cases} \frac{1}{\lambda_n} [\mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{x}^*)]_n = \phi'(x_n), & x_n^* \neq 0 \\ \frac{1}{\lambda_n} [\mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{x}^*)]_n \in [\phi(0^-), \phi(0^+)], & x_n^* = 0 \end{cases} \quad (45)$$

where $[\mathbf{v}]_n$ denotes the n -th component of the vector \mathbf{v} .

In the example below, the optimality of the numerically obtained solution is illustrated by a scatter plot of $[\mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{x})]_n / \lambda_n$ versus $x_n a_n$, for $n \in \mathbb{Z}_N$, in Fig. 7.

The condition (45) can be used to set the threshold value for the penalty function $\phi(x)$. Suppose \mathbf{y} follows the model (35) where \mathbf{x} is sparse. One approach for setting λ_n is to ask that the solution to (36) be all-zero when \mathbf{x} is all-zero in the model (35). Note that, if $\mathbf{x} = \mathbf{0}$, then \mathbf{y} consists of noise only (i.e., $\mathbf{y} = \mathbf{w}$). In this case, (45) suggests that λ_n be chosen such that

$$\lambda_n \phi(0^-) \leq [\mathbf{H}^T \mathbf{w}]_n \leq \lambda_n \phi(0^+), \quad n \in \mathbb{Z}_N. \quad (46)$$

For the ℓ_1 norm, logarithmic and arctangent penalty functions, $\phi(0^-) = -1$ and $\phi(0^+) = 1$, so (46) can be written as

$$|[\mathbf{H}^T \mathbf{w}]_n| \leq \lambda_n, \quad n \in \mathbb{Z}_N. \quad (47)$$

However, the larger λ_n is, the more x_n will be attenuated. Hence, it is reasonable to set λ_n to the smallest value satisfying (47), namely,

$$\lambda_n \approx \max |[\mathbf{H}^T \mathbf{w}]_n| \quad (48)$$

where \mathbf{w} is the additive noise. Although (48) assumes availability of the noise signal \mathbf{w} , which is unknown in practice, (48) can often be estimated based on knowledge of statistics of the noise \mathbf{w} . For example, based on the ‘three-sigma rule’, we obtain

$$\lambda_n \approx 3 \text{std}([\mathbf{H}^T \mathbf{w}]_n). \quad (49)$$

If \mathbf{w} is white Gaussian noise with variance σ^2 , then

$$\text{std}([\mathbf{H}^T \mathbf{w}]_n) = \sigma \|\mathbf{H}(\cdot, n)\|_2 \quad (50)$$

where $\mathbf{H}(\cdot, n)$ denotes column n of \mathbf{H} . For example, if \mathbf{H} denotes linear convolution, then all columns of \mathbf{H} have equal norm and (49) becomes

$$\lambda_n = \lambda \approx 3\sigma \|\mathbf{h}\|_2 \quad (51)$$

where \mathbf{h} is the impulse of the convolution system.

D. Usage of Method

We summarize the forgoing approach, MSC, to sparsity penalized least squares, cf. (36). We assume the parameters λ_n are fixed (e.g., set according to additive noise variance).

- 1) Input: $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{H} \in \mathbb{R}^{M \times N}$, $\{\lambda_n > 0, n \in \mathbb{Z}_N\}$, $\phi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.
- 2) Find a positive semidefinite diagonal matrix \mathbf{R} such that $\mathbf{H}^T \mathbf{H} - \mathbf{R}$ is positive semidefinite; i.e., solve (44), or use the sub-optimal $\mathbf{R} = \alpha_{\min} \mathbf{I}$. Denote the diagonal elements of \mathbf{R} by $r_n, n \in \mathbb{Z}_N$.
- 3) For $n \in \mathbb{Z}_N$, set a_n such that $(r_n/\lambda_n, a_n) \in \mathcal{S}$. Here, \mathcal{S} is the set such that f in (37) is convex if $(\lambda, a) \in \mathcal{S}$.
- 4) Minimize (36) to obtain \mathbf{x} .
- 5) Output: $\mathbf{x} \in \mathbb{R}^N$. \square

The penalty function ϕ need not be the logarithmic or arctangent penalty functions discussed above. Another parametric penalty function can be used, but it must have the property that f in (37) is convex for $(\lambda, a) \in \mathcal{S}$ for some set \mathcal{S} . Note that $\phi(x, p) = |x|^p$ with $0 < p < 1$ does not qualify because f is non-convex for all $0 < p < 1$. On the other hand, the firm penalty function [28] could be used.

In step (3), for the logarithmic and arctangent penalty functions, one can use

$$a_n = \beta \frac{r_n}{\lambda_n}, \quad \text{where } 0 \leq \beta \leq 1. \quad (52)$$

When $\beta = 0$, the penalty function is simply the ℓ_1 norm; in this case, the proposed method offers no advantage relative to ℓ_1 norm penalized least squares (BPD/lasso). When $\beta = 1$, the penalty function is maximally non-convex (maximally sparsity-inducing) subject to F being convex. We have used $\beta = 0.9$ in the examples below.

The minimization of (36) in step (4) is a convex optimization problem for which numerous algorithms have been developed as noted in Sec. I-B. The most efficient algorithm depends primarily on the properties of \mathbf{H} .

E. Iterative MSC (IMSC)

An apparent limitation of the proposed approach, MSC, is that for some problems of interest, the parameters r_n are either equal to zero or nearly equal to zero for all $n \in \mathbb{Z}_N$, i.e., $\mathbf{R} \approx \mathbf{0}$. In this case, the method requires that $\phi(\cdot; a_n)$ be convex or practically convex. For example, for the logarithmic and arctangent penalty functions, $r_n \approx 0$ leads to $a_n \approx 0$. As a consequence, the penalty function is practically the ℓ_1 norm. In this case, the method offers no advantage in comparison with ℓ_1 norm penalized least squares (BPD/lasso).

The situation wherein $\mathbf{R} \approx \mathbf{0}$ arises in two standard sparse signal processing problems: basis pursuit denoising and deconvolution. In deconvolution, if the system is non-invertible or nearly singular (i.e., the frequency response has a null or approximate null at one or more frequencies), then the optimal lower bound \mathbf{R} will be $\mathbf{R} \approx \mathbf{0}$. In BPD, the matrix \mathbf{H} often represents the inverse of an overcomplete frame (or dictionary), in which case the optimal lower bound \mathbf{R} is again close to zero.

In order to broaden the applicability of MSC, we describe iterative MSC (IMSC) wherein MSC is applied several times.

On each iteration, MSC is applied only to the non-zero elements of the sparse solution \mathbf{x} obtained as a result of the previous iteration. Each iteration involves only those columns of \mathbf{H} corresponding the previously identified non-zero components. As the number of active columns of \mathbf{H} diminishes as the iterations progress, the problem (44) produces a sequence of increasingly positive diagonal matrix \mathbf{R} . Hence, as the iterations progress, the penalty functions become increasingly non-convex. The procedure can be repeated until there is no change in the index set of non-zero elements.

The IMSC algorithm is initialized with the ℓ_1 norm solution, i.e., using $\phi(x, a_n) = |x|$ for all $n \in \mathbb{Z}_N$. (For the logarithmic and arctangent penalties, $a_n = 0, n \in \mathbb{Z}_N$.) We assume the ℓ_1 norm solution is reasonably sparse; otherwise, sparsity is likely not useful for the problem at hand. The algorithm should be terminated when there is no change (or only insignificant change) between the active set from one iteration to the next.

The IMSC procedure is described as follows, where $i \geq 1$ denotes the iteration index.

- 1) Initialization. Find the ℓ_1 norm solution:

$$\mathbf{x}^{(1)} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sum_{n=0}^{N-1} \lambda_n |x_n|. \quad (53)$$

Set $i = 1$ and $K^{(0)} = N$. Note \mathbf{H} is of size $M \times N$.

- 2) Identify the non-zero elements of $\mathbf{x}^{(i)}$, and record their indices in the set $\mathcal{K}^{(i)}$,

$$\mathcal{K}^{(i)} = \left\{ n \in \mathbb{Z}_N \mid x_n^{(i)} \neq 0 \right\}. \quad (54)$$

This is the support of $\mathbf{x}^{(i)}$. Let $K^{(i)}$ be the number of non-zero elements of $\mathbf{x}^{(i)}$, i.e., $K^{(i)} = |\mathcal{K}^{(i)}|$.

- 3) Check the termination condition: If $K^{(i)}$ is not less than $K^{(i-1)}$, then terminate. The output is $\mathbf{x}^{(i)}$.
- 4) Define $\mathbf{H}^{(i)}$ as the sub-matrix of \mathbf{H} containing only columns $k \in \mathcal{K}^{(i)}$. The matrix $\mathbf{H}^{(i)}$ is of size $M \times K^{(i)}$. Find a positive semidefinite diagonal matrix $\mathbf{R}^{(i)}$ lower bounding $[\mathbf{H}^{(i)}]^T \mathbf{H}^{(i)}$, i.e., solve problem (44) or use $\alpha_{\min}^{(i)} \mathbf{I}$. $\mathbf{R}^{(i)}$ is of size $K^{(i)} \times K^{(i)}$.
- 5) Set a_n such that $(\lambda_n/r_n^{(i)}, a_n) \in \mathcal{S}$, $n \in \mathcal{K}^{(i)}$. For example, with the logarithmic and arctangent penalties, one may set

$$a_n^{(i)} = \beta \frac{r_n^{(i)}}{\lambda_n}, \quad n \in \mathcal{K}^{(i)} \quad (55)$$

for some $0 \leq \beta \leq 1$.

- 6) Solve the $K^{(i)}$ dimensional convex problem:

$$\mathbf{u}^{(i)} = \arg \min_{\mathbf{u} \in \mathbb{R}^{K^{(i)}}} \|\mathbf{y} - \mathbf{H}^{(i)}\mathbf{u}\|_2^2 + \sum_{n \in \mathcal{K}^{(i)}} \lambda_n \phi(u_n; a_n^{(i)}). \quad (56)$$

- 7) Set $\mathbf{x}^{(i+1)}$ as

$$x_n^{(i+1)} = \begin{cases} 0, & n \notin \mathcal{K}^{(i)} \\ u_n^{(i)}, & n \in \mathcal{K}^{(i)}. \end{cases} \quad (57)$$

- 8) Set $i = i + 1$ and go to step 2). \square

In the IMSC algorithm, the support of $\mathbf{x}^{(i)}$ can only shrink from one iteration to the next, i.e., $\mathcal{K}^{(i+1)} \subseteq \mathcal{K}^{(i)}$ and $K^{(i+1)} \leq K^{(i)}$. Once there is no further change in $\mathcal{K}^{(i)}$, each subsequent iteration will produce exactly the same result, i.e.,

$$\mathcal{K}^{(i+1)} = \mathcal{K}^{(i)} \implies \mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}. \quad (58)$$

For this reason, the procedure should be terminated when $\mathcal{K}^{(i)}$ ceases to shrink. In the 1D sparse deconvolution example below, the IMSC procedure terminates after only three or four iterations.

Note that the size of problem (44) in step 4) reduces in size as the algorithm progresses. Hence each instance of (44) requires less computation than the previous. More importantly, each matrix $\mathbf{H}^{(i+1)}$ has a subset of the columns of $\mathbf{H}^{(i)}$. Hence, $\mathbf{R}^{(i+1)}$ is less constrained than $\mathbf{R}^{(i)}$, and the penalty functions become more non-convex (more strongly sparsity-inducing) as the iterations progress. Therefore, the IMSC algorithm produces a sequence of successively sparser $\mathbf{x}^{(i)}$.

Initializing the IMSC procedure with the ℓ_1 norm solution substantially reduces the computational cost of the algorithm. Note that if the ℓ_1 norm solution is sparse, i.e., $K^{(1)} \ll N$, then all the semidefinite optimization problems (44) have far fewer variables than N , i.e., $K^{(i)} \leq K^{(1)}$. Hence, IMSC can be applied to larger data sets than would otherwise be computationally practical, given the computational cost of (44).

F. Deconvolution Example

A sparse signal $x(n)$ of length $N = 1000$ is generated so that (i) the inter-spike interval is uniform random between 5 and 35 samples, and (ii) the amplitude of each spike is uniform between -1 and 1 . The signal is illustrated in Fig. 6.

The spike signal is then used as the input to a linear time-invariant (LTI) system, the output of which is contaminated by AWGN, $w(n)$. The observed data, $y(n)$, is written as

$$y(n) = \sum_k b(k) x(n-k) - \sum_k a(k) y(n-k) + w(n)$$

where $w(n) \sim \mathcal{N}(0, \sigma^2)$. It can also be written as

$$\mathbf{y} = \mathbf{A}^{-1} \mathbf{B} \mathbf{x} + \mathbf{w} = \mathbf{H} \mathbf{x} + \mathbf{w}, \quad \mathbf{H} = \mathbf{A}^{-1} \mathbf{B}$$

where \mathbf{A} and \mathbf{B} are banded Toeplitz matrices [50]. In this example, we set $b(0) = 1$, $b(1) = 0.8$, $a(0) = 1$, $a(1) = -1.047$, $a(2) = 0.81$, and $\sigma = 0.2$. The observed data, \mathbf{y} , is illustrated in Fig. 6.

Several algorithms for estimating the sparse signal \mathbf{x} will be compared. The estimated signal is denoted $\hat{\mathbf{x}}$. The accuracy of the estimation is quantified by the ℓ_2 and ℓ_1 norms of the error signal and by the support error, denoted L2E, L1E, and SE respectively.

- 1) $\text{L2E} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2$
- 2) $\text{L1E} = \|\mathbf{x} - \hat{\mathbf{x}}\|_1$
- 3) $\text{SE} = \|s(\mathbf{x}) - s(\hat{\mathbf{x}})\|_0$

The support error, SE, is computed using $s(\mathbf{x})$, the ϵ -support of $\mathbf{x} \in \mathbb{R}^N$. Namely, $s : \mathbb{R}^N \rightarrow \{0, 1\}^N$ is defined as

$$[s(\mathbf{x})]_n = \begin{cases} 1, & |x_n| > \epsilon \\ 0, & |x_n| \leq \epsilon \end{cases} \quad (59)$$

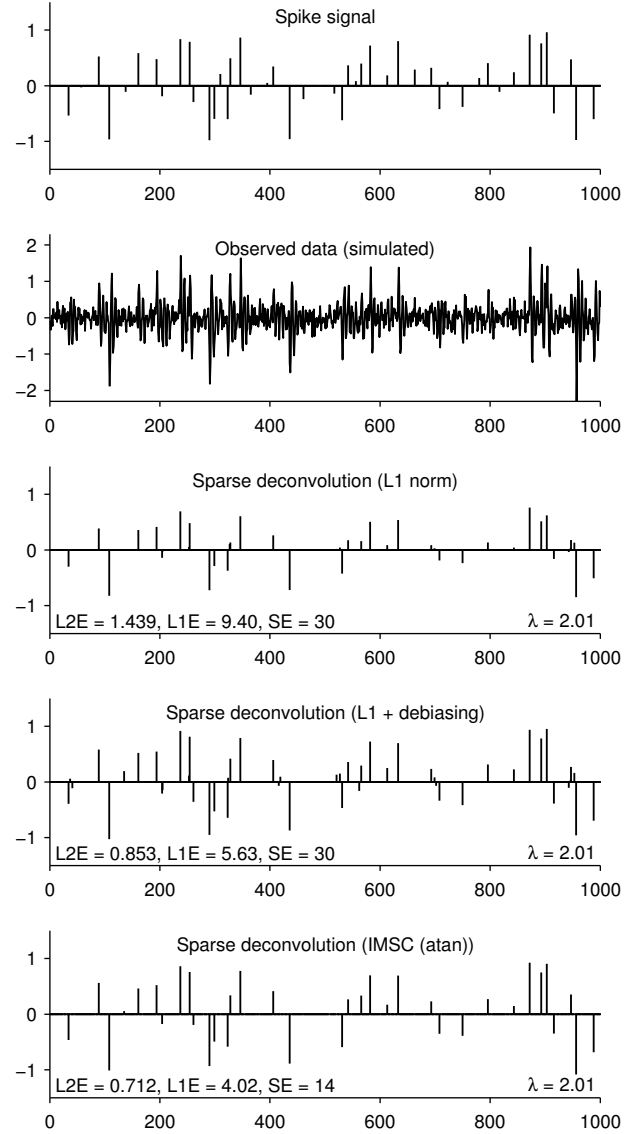


Fig. 6. Sparse deconvolution via sparsity penalized least squares.

where $\epsilon > 0$ is a small value to accommodate negligible non-zeros. We set $\epsilon = 10^{-3}$. The support error, SE, counts both the false zeros and the false non-zeros of $\hat{\mathbf{x}}$. The numbers of false zeros and false non-zeros are denoted FZ and FN, respectively.

First, the sparse ℓ_1 norm solutions, i.e., $\phi(x, a) = |x|$ in (36), with and without debiasing, are computed. We set λ_n according to (51), i.e., $\lambda_n = 2.01$, $n \in \mathbb{Z}_N$. The estimated signals are illustrated in Fig. 6. The errors L2E, L1E, and SE, are noted in the figure. As expected, debiasing substantially improves the L2E and L1E errors of the ℓ_1 norm solution; however, it does not improve the support error, SE. In particular, debiasing does not make the solution more sparse. The errors, averaged over 200 trials, are shown in Table I. Each trial consists of independently generated sparse and noise signals.

Next, sparse deconvolution is performed using three algorithms developed to solve the highly non-convex ℓ_0 quasi-norm problem, namely the Iterative Support Detection (ISD)

TABLE I
SPARSE DECONVOLUTION EXAMPLE. AVERAGE ERRORS (200 TRIALS).

Algorithm	L2E	L1E	SE	(FZ, FN)
ℓ_1 norm	1.443	10.01	37.60	(10.3, 27.3)
ℓ_1 norm + debiasing	0.989	7.14	37.57	(10.3, 27.2)
AIHT [6]	1.073	6.37	24.90	(12.4, 12.5)
ISD [56]	0.911	5.19	19.67	(11.6, 8.1)
SBR [51]	0.788	4.05	13.62	(12.0, 1.6)
ℓ_p ($p = 0.7$) IRL2	0.993	5.80	16.32	(12.9, 3.4)
ℓ_p ($p = 0.7$) IRL2 + debiasing	0.924	4.82	16.32	(12.9, 3.4)
ℓ_p ($p = 0.7$) IRL1	0.884	5.29	14.43	(11.5, 2.9)
ℓ_p ($p = 0.7$) IRL1 + debiasing	0.774	4.18	14.43	(11.5, 2.9)
IMSC (log)	0.896	5.35	19.50	(10.4, 9.1)
IMSC (log) + debiasing	0.834	5.00	19.50	(10.4, 9.1)
IMSC (atan)	0.794	4.52	16.93	(10.5, 6.4)
IMSC (atan) + debiasing	0.790	4.54	16.92	(10.5, 6.4)
IMSC/S (atan)	0.967	5.94	20.16	(10.3, 9.9)
IMSC/S (atan) + debiasing	0.826	5.02	20.14	(10.3, 9.9)

algorithm [56],² the Accelerated Iterative Hard Thresholding (AIHT) algorithm [6],³ and the Single Best Replacement (SBR) algorithm [51]. In each case, we used software by the respective authors. The ISD and SBR algorithms require regularization parameters ρ and λ respectively; we found that $\rho = 1.0$ and $\lambda = 0.5$ were approximately optimal. The AIHT algorithm requires the number of non-zeros be specified; we used the number of non-zeros in the true sparse signal. Each of ISD, AIHT, and SBR significantly improve the accuracy of the result in comparison with the ℓ_1 norm solutions, with SBR being the most accurate. These algorithms essentially seek the correct support. They do not penalize the values in the detected support; so, debiasing does not alter the signals produced by these algorithms.

The ℓ_p quasi-norm, with $p = 0.7$, i.e. $\phi(x) = |x|^p$, also substantially improves upon the ℓ_1 norm result. Several methods exist to minimize the cost function F in this case. We implement two methods: IRL2 and IRL1 (iterative reweighted ℓ_2 and ℓ_1 norm minimization, respectively), with and without debiasing in each case. We used $\lambda = 1.0$, which we found to be about optimal on average for this deconvolution problem. As revealed in Table 6, IRL1 is more accurate than IRL2. Note that IRL2 and IRL1 seek to minimize exactly the same cost function; so the inferiority of IRL2 compared to IRL1 is due to the convergence of IRL2 to a local minimizer of F . Also note that debiasing substantially improves L2E and L1E (with no effect on SE) for both IRL2 and IRL1. The ℓ_p results demonstrate both the value of a non-convex regularizer and the vulnerability of non-convex optimization to local minimizers.

The results of the proposed iterative MSC (IMSC) algorithm, with and without debiasing, are shown in Table I. We used $\beta = 0.9$ and $\lambda_n = 2.01$, $n \in \mathbb{Z}_N$, in accordance with (51). Results using the logarithmic (log) and arctangent (atan) penalty functions are tabulated, which show the improvement provided by the later penalty, in terms of L2E, L1E, and SE. While debiasing reduces the error (bias) of the logarithmic penalty, it has negligible effect on the arctangent penalty. The simplified form of the MSC algorithm, wherein $\mathbf{R} = \alpha_{\min} \mathbf{I}$ is used instead of the optimal \mathbf{R} computed using SDP, is also

tabulated in Table I, denoted by IMSC/S. IMSC/S is more computationally efficient than MSC due to the omission of SDP; however, it does lead to an increase in the error measures.

The IMSC algorithm ran for three iterations on average. For example, the IMSC solution illustrated in Fig. 6 ran with $K^{(1)} = 61$, $K^{(2)} = 40$, and $K^{(3)} = 38$. Therefore, even though the signal is of length 1000, the SDPs that had to be solved are much smaller: of sizes 61, 40, and 38, only.

The optimality of the MSC solution at each stage can be verified using (45). Specifically, a scatter plot of $[\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x})]_n / \lambda_n$ versus $x_n a_n$, for all $n \in \mathcal{K}^{(i)}$, should show all points lying on the graph of $\partial\phi(x, 1)$. For the IMSC solution illustrated in Fig. 6, this optimality scatter plot is illustrated in Fig. 7, which shows that all points lie on $\text{sign}(x)/(1+|x|+x^2)$, hence verifying the optimality of the obtained solution.

To more clearly compare the relative bias of the ℓ_1 norm and IMSC (atan) solutions, they are illustrated together in Fig. 7. Only the non-zero elements of each solution are shown. In this figure, the closer the points lie to the identity, the more accurate the solution. The figure shows that the ℓ_1 norm solution systematically underestimates the true values more so than does the MSC (atan) solution.

In terms of L2E and L1E, the best IMSC result, i.e., IMSC (atan), is outperformed by SBR and the IRL1 + debiasing algorithm. In addition, IMSC (atan) yields lower SE than ℓ_1 minimization, AIHT, and ISD. IMSC does not yield the best error measures, but it comes reasonably close; even though, IMSC is based entirely on convex optimization. In terms of L1E and SE, the SBR performs best for this example. Most notably, SBR attains a small number of false non-zeros.

Note that IMSC requires only the parameter β (with $0 \leq \beta \leq 1$) beyond those parameters (namely λ_n) required for the ℓ_1 norm solution.

Fig. 8 illustrates the average errors as functions of the regularization parameter, for ISD, IMSC, and IMSC + debiasing (denoted IMSC+d in the figure). For IMSC, the regularization parameter is λ . For ISD, the regularization parameter is $\rho = \lambda/2$. Note that for IMSC, the value of λ minimizing L2E and L1E depends on whether or not debiasing is performed. The value λ suggested by (51) (i.e., $\lambda = 2$) is reasonably effective with or without debiasing. The value of λ minimizing SE is somewhat higher.

The implementation of the ℓ_1 , IRL2, IRL1, and IMSC algorithms for deconvolution each require the solution of (36) with various penalty functions and/or sub-matrices of \mathbf{H} . We have used algorithms, based on majorization of the penalty function, that exploit banded matrix structures for computational efficiencies [49], [50].

IV. CONCLUSION

This paper proposes an approach (MSC) to obtain sparse solutions to ill-posed linear inverse problems. In order to induce sparsity more strongly than the ℓ_1 norm, the MSC approach utilizes non-convex penalty functions. However, the non-convex penalty functions are constrained so that the total cost function is convex. The maximally non-convex (maximally sparsity-inducing) constrained penalty functions

²<http://www.caam.rice.edu/%7Eoptimization/L1/ISD/>

³<http://users.fmrib.ox.ac.uk/%7Etblumens/sparsify/sparsify.html>

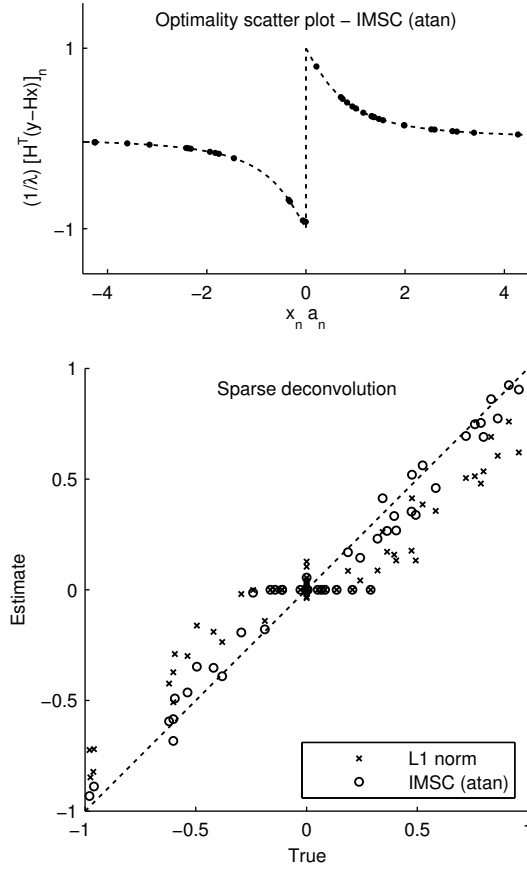


Fig. 7. Sparse deconvolution. (a) Illustration of optimality condition (45) for IMSC (atan) solution. (b) Comparison of ℓ_1 norm and IMSC solutions.

are found by formulating a semidefinite program (SDP). Iterative MSC (IMSC) consists of applying MSC to the non-zero (active) elements of the sparse solution produced by the previous iteration.

The MSC method is intended as a convex alternative to ℓ_1 norm minimization, which is widely used in sparse signal processing. Being based entirely on convex optimization, it can not be expected that MSC produces solutions as sparse as non-convex optimization methods, such as ℓ_p quasi-norm ($0 < p < 1$) minimization. However, it provides a principled approach for enhanced sparsity relative to the ℓ_1 norm. Moreover, although it is not explored here, it may be effective to use MSC in conjunction with other techniques. As has been recognized and illustrated in the sparse deconvolution example above, reweighted ℓ_1 norm minimization can be more effective than reweighted ℓ_2 norm minimization (i.e., higher likelihood of convergence to a global minimizer). Likewise, it will be of interest to explore the use of reweighted MSC or similar methods as a means of more reliable non-convex optimization. For example, a non-convex algorithm of the MM-type may be devised wherein a general non-convex penalty function is majorized by a non-convex function constrained so as to ensure convexity of the total cost function.

To apply the proposed approach to large scale problems in practice, it will be necessary to have an algorithm for solving (44) which does not rely on accessing or manipulating individual rows or columns of \mathbf{H} .

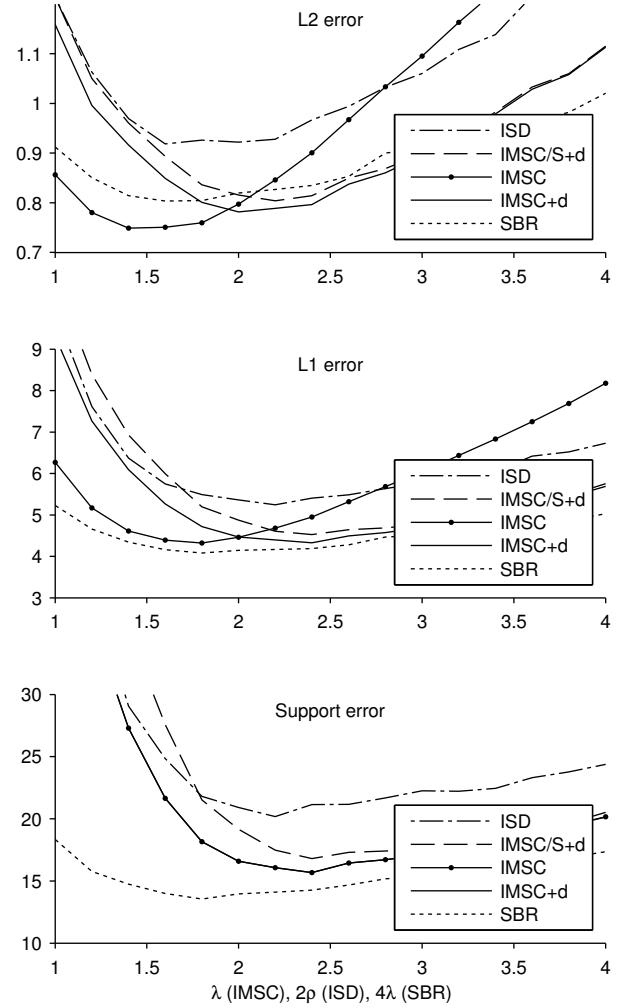


Fig. 8. Sparse deconvolution. Errors as functions of regularization parameters, averaged over 50 realizations. (Note that the support error for IMSC and IMSC+d coincide.)

REFERENCES

- [1] A. Achim, P. Tsakalides, and A. Bezerianos. SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling. *IEEE Trans. on Geoscience and Remote Sensing*, 41(8):1773–1784, August 2003.
- [2] A. Antoniou and W. S. Lu. *Practical Optimization: Algorithms and Engineering Applications*. Springer, 2007.
- [3] A. M. Atto, D. Pastor, and G. Mercier. Wavelet shrinkage: unification of basic thresholding functions and thresholds. *Signal, Image and Video Proc.*, 5:11–28, 2011.
- [4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.
- [5] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma, editors. Special issue on applications of sparse representation and compressive sensing. *Proc. IEEE*, 98(6), June 2010.
- [6] A. Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.
- [7] T. Blumensath and M. E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE. J. Sel. Top. Signal Processing*, 4(2):298–309, April 2010.
- [8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [9] M. S. O’ Brien, A. N. Sinclair, and S. M. Kramer. Recovery of a

- sparse spike time series by L1 norm deconvolution. *IEEE Trans. Signal Process.*, 42(12):3353–3365, December 1994.
- [10] E. J. Candès, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.*, 14(5):877–905, December 2008.
 - [11] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.*, 6(2):298–311, February 1997.
 - [12] R. Chartrand. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. In *IEEE Int. Symp. Biomed. Imag. (ISBI)*, pages 262–265, July 2009.
 - [13] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.
 - [14] J. F. Claerbout and F. Muir. Robust modeling of erratic data. *Geophysics*, 38(5):826–844, 1973.
 - [15] M. Clyde and E. I. George. Empirical Bayes estimation in wavelet nonparametric regression. In P. Muller and B. Vidakovic, editors, *Bayesian Inference in Wavelet Based Models*, pages 309–322. Springer-Verlag, 1999.
 - [16] P. L. Combettes and J.-C. Pesquet. Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM J. Optim.*, 18(4):1351–1376, 2008.
 - [17] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke et al., editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, 2010.
 - [18] D. Donoho, A. Maleki, and M. Shahrar. Wavelab 850. <http://www-stat.stanford.edu/%7Ewavelab/>.
 - [19] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
 - [20] E. Esser, X. Zhang, and T. F. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.*, 3(4):1015–1046, 2010.
 - [21] J. M. Fadili and L. Boubchir. Analytical form for a Bayesian wavelet estimator of images using the Bessel K form densities. *IEEE Trans. Image Process.*, 14(2):231–240, February 2005.
 - [22] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360, 2001.
 - [23] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16(12):2980–2991, December 2007.
 - [24] M. Figueiredo and R. Nowak. Wavelet-based image estimation: An empirical Bayes approach using Jeffrey’s noninformative prior. *IEEE Trans. Image Process.*, 10(9):1322–1331, September 2001.
 - [25] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.*, 1(4):586–598, December 2007.
 - [26] S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM J. Numer. Anal.*, 49(6):2543–2563, 2010.
 - [27] H. Gao. Wavelet shrinkage denoising using the nonnegative garrote. *J. Comput. Graph. Statist.*, 7:469–488, 1998.
 - [28] H.-Y. Gao and A. G. Bruce. Waveshrink with firm shrinkage. *Statistica Sinica*, 7:855–874, 1997.
 - [29] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Trans. Signal Process.*, 57(12):4686–4698, December 2009.
 - [30] A. Gholami and S. M. Hosseini. A general framework for sparsity-based denoising and inversion. *IEEE Trans. Signal Process.*, 59(11):5202–5211, November 2011.
 - [31] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imag. Sci.*, 2(2):323–343, 2009.
 - [32] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.*, 45(3):600–616, March 1997.
 - [33] G. Hari Kumar and Y. Bresler. A new algorithm for computing sparse solutions to linear inverse problems. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 3, pages 1331–1334, May 1996.
 - [34] A. Hyvärinen. Sparse code shrinkage: Denoising of non-Gaussian data by maximum likelihood estimation. *Neural Computation*, 11:1739–1768, 1999.
 - [35] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *IEEE Trans. Signal Process.*, 56(6):2346–2356, June 2008.
 - [36] K. F. Kaarensen. Deconvolution of sparse spike trains by iterated window maximization. *IEEE Trans. Signal Process.*, 45(5):1173–1183, May 1997.
 - [37] N. Kingsbury and T. Reeves. Redundant representation with complex wavelets: how to achieve sparsity. In *Proc. IEEE Int. Conf. Image Processing*, 2003.
 - [38] I. Kozlov and A. Petukhov. Sparse solutions of underdetermined linear systems. In W. Freeden et al., editor, *Handbook of Geomathematics*. Springer, 2010.
 - [39] D. A. Lorenz. Non-convex variational denoising of images: Interpolation between hard and soft wavelet shrinkage. *Current Development in Theory and Application of Wavelets*, 1(1):31–56, 2007.
 - [40] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.
 - [41] H. Mohimani, M. Babaie-Zadeh, and C. Jutten. A fast approach for overcomplete sparse decomposition based on smoothed l0 norm. *IEEE Trans. Signal Process.*, 57(1):289–301, January 2009.
 - [42] N. Mourad and J. P. Reilly. Minimizing nonconvex functions for sparse vector reconstruction. *IEEE Trans. Signal Process.*, 58(7):3485–3496, July 2010.
 - [43] S. Nadarajah and S. Kotz. The BKF Bayesian wavelet estimator. *Signal Processing*, 87(9):2268–2271, September 2007.
 - [44] J. Portilla and L. Mancera. L0-based sparse approximation: two alternative methods and some applications. In *Proceedings of SPIE*, volume 6701 (Wavelets XII), 2007.
 - [45] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Process.*, 12(11):1338–1351, November 2003.
 - [46] K. Qiu and A. Dogandzic. Sparse signal reconstruction via ECME hard thresholding. *IEEE Trans. Signal Process.*, 60(9):4551–4569, September 2012.
 - [47] I. Ramirez and G. Sapiro. Universal regularizers for robust sparse coding and modeling. *IEEE Trans. Image Process.*, 21(9):3850–3864, September 2012.
 - [48] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE Trans. Signal Process.*, 51(3):760–770, March 2003.
 - [49] I. Selesnick. Penalty and shrinkage functions for sparse signal processing. *Connexions Web site*, 2012. <http://cnx.org/content/m45134/1.1/>.
 - [50] I. Selesnick. Sparse deconvolution (an MM algorithm). *Connexions Web site*, 2012. <http://cnx.org/content/m44991/1.4/>.
 - [51] C. Soussen, J. Idier, D. Brie, and J. Duan. From Bernoulli-Gaussian deconvolution to sparse signal restoration. *IEEE Trans. Signal Process.*, 59(10):4572–4584, October 2011.
 - [52] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from <http://fewcal.kub.nl/sturm>.
 - [53] X. Tan, W. Roberts, J. Li, and P. Stoica. Sparse learning via iterative minimization with application to MIMO radar imaging. *IEEE Trans. Signal Process.*, 59(3):1088–1101, March 2011.
 - [54] H. L. Taylor, S. C. Banks, and J. F. McCoy. Deconvolution with the l1 norm. *Geophysics*, 44(1):39–52, 1979.
 - [55] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B*, 58(1):267–288, 1996.
 - [56] Y. Wang and W. Yin. Sparse signal reconstruction via iterative support detection. *SIAM J. Imag. Sci.*, 3(3):462–491, 2010.
 - [57] D. Wipf and S. Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE J. Sel. Top. Signal Processing*, 4(2):317–329, April 2010.
 - [58] X.-P. Zhang. Thresholding neural network for adaptive noise reduction. *IEEE Trans. Neural Networks*, 12(3):567–584, May 2001.
 - [59] X.-P. Zhang and M. D. Desai. Adaptive denoising based on SURE risk. *IEEE Signal Processing Letters*, 5(10):265–267, October 1998.
 - [60] Z.-D. Zhao. Wavelet shrinkage denoising by generalized threshold function. In *Proc. Int. Conf. Machine Learning and Cybernetics*, volume 9, pages 5501–5506, August 2005.
 - [61] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Statist.*, 36(4):1509–1533, 2008.